R E P O R T R E S U M E S

ED 003 801 64

WHAT IS PROGRAMED INSTRUCTION, AN INTRODUCTION FOR THE LAYMAN.

BY- SCHRAMM, WILBUR
STANFORD UNIV., CALIF., INST.FOR COMMUNIC. RES.

REPORT NUMBER NDEA-VIIB-322-2 PUB DATE 64
EDRS PRICE MF-$0.09 HC-$1.08 27P.

DESCRIPTORS- *PROGRAMED INSTRUCTION, *RESOURCE MATERIALS, *PUBLICATIONS, *PROGRAMING, *RESOURCE GUIDES, INFORMATION DISSEMINATION, TEACHING MACHINES, BIBLIOGRAPHY, STANFORD, CALIFORNIA

VARIOUS ASPECTS OF PROGRAMED INSTRUCTION WERE PRESENTED TO INTRODUCE THIS SUBJECT TO A LAY AUDIENCE. THE TOPICS DISCUSSED INCLUDED--(1) ESSENTIALS OF PROGRAMED INSTRUCTION, (2) TRENDS IN PROGRAMED INSTRUCTION, (3) DEVELOPMENT OF PROGRAMS, (4) TYPES OF PROGRAMS, (5) TEACHING MATERIAL WHICH CAN BE PROGRAMED, (6) EFFECTIVENESS OF PROGRAMS, (7) EVALUATION OF PROGRAMS, AND (8) UTILITY OF TEACHING MACHINES. (RS)

STANFORD *Institute for Communication Research*

No2

U. S. DEPARTMENT OF HEALTH, EDUCATION AND WELFARE
Office of Education
This document has been reproduced exactly as received from the person or organization originating it. Points of view or opinions stated do not necessarily represent official Office of Education position or policy.

WHAT IS PROGRAMED INSTRUCTION?

An introduction for the layman

Wilbur Schramm

ED002801

This report is one of a number prepared in the Institute for Communication Research, 1964, Stanford University, pursuant to a contract with the United States Office of Education, Department of Health, Education, and Welfare.

The essentials of programed instruction

A good way to understand how programed instruction works is to recall, from your school years, a really excellent tutorial session.

I can remember one such session that was intended to help me learn to add fractions. As nearly as I can remember, this is what happened.

The tutor explored a bit at first. He wanted to find out how much I already knew about fractions, so he would know where to start teaching. When he found that starting point, he started at once to lead me into areas of knowledge where I had never been.

Most of the hour was questions and answers. At first he asked one or two things I knew well. Then came a question that was new to me, but one that I could answer from having answered the preceding ones. And then another question that followed rather easily after <u>that</u> one. Thus he led me, step by step, through the processes he wanted me to learn. Each step was so easy, and followed so logically on what had gone before, that I could usually manage the step without slipping or needing explanation. If I did make an error, he told me; and I wasn't allowed to go ahead until I had given the right answer. Often when I made an error he would ask additional questions so that I could see where I had been wrong. Occasionally he explained. Frequently he reviewed. All this in a low-key, friendly, conversational way. He gave me harder and harder problems, and less and less help with them. And finally when he was satisfied that I could perform the additions he thought I should be able to, he patted me on the shoulder and dismissed me.

Even at the time, and despite my schoolboy impatience with teachers and school, still I appreciated the great skill of that kind of teaching. I also realized how different it was from most of the learning experiences I had in school. Unlike a lecture or a textbook, that experience didn't tell me: it kept me active, made me find out for myself. On the other hand, it didn't let me flounder around: rather, it gave me a map of the shortest way from what I knew to what I needed to learn, and told me at every step whether I was still on the path. Unlike a film or a lecture, it was designed to fit me: it started where I was, and let me go as fast or as slowly as I needed to. And it was rather satisfying because I was moving toward what I later learned to call a "behavioral objective": I was learning to do a specified thing. At the beginning of it I couldn't add fractions; at the end, I could. I could look back with a sense of accomplishment, and ahead (at a world apparently full of fractions) with some confidence.

I remember thinking that I wished more of my schoolwork was like that. But it was many years before I realized that what I had experienced that day was something very much like programed instruction.

Modern programed instruction

The principal difference between my tutorial session and modern programed instruction is that a program -- printed or photographed or fed into a computer -- can now be used to sit in for the tutor, reproduce the experience over and over again for other students, and thus enormously expand the opportunity for this kind of learning. It doesn't much matter

whether the program is in a teaching machine or in the form of a programed text or on a tape in a computer. It doesn't much matter whether a student responds to the program by writing or speaking answers, selecting multiple-choice answers, pushing buttons, typing an answer on a computer-connected typewriter, or whatever. Make no mistake about it -- what matters is the program! And the program does essentially what the tutor did in the session I have just described.

Here is an example -- a very brief program written by Basic Systems, Inc., and reproduced with their permission:*

---

*This demonstration program is found in a useful brochure, Introduction to Programmed Instruction (New York: Basic Systems, Inc., 1962).

QUESTIONS

1. Here is how you can multiply 45 x 45.

   Between what two multiples of ten does 45 lie? Obviously 40 and 50. Multiply these two numbers:

   40 x ___ = 2000

   Add 25 to the product:

   ___ + ___ = 2025

ANSWERS

1. If you want to check this, multiply 45 x 45 the long way.

2. Here is how you can multiply 75 x 75.

   Between what two multiples of ten does 75 lie? 70 and ___

   Multiply these two numbers:

   ___ x 80 = _____

   Add 25 to the product: _____

2. Your answer should be 5625

From now on until the end of the program, any time you feel you can multiply 95 x 95, or 55 x 55 <u>mentally</u>, try question 6. Otherwise, go on to the next question.

3. Here is how you can multiply 35 x 35. Do the steps <u>mentally</u>.

   Between which two multiples of 10 does 35 lie? ___ and 40.

   Multiply these: _____

   Add 25: _____

3. Your answer should be 1225

4. Here is how you can multiply 85 x 85. Do the steps <u>mentally</u>.

   Between which two multiples of 10 . . . _____ and _____

   Multiply . . . _____

   . . . . . _____

4. Your answer should be 7225

QUESTIONS	ANSWERS
5. Here is how you can multiply 15 x 15. Do the steps mentally. Between which . . . _____ . . . . . _____ . . . . . _____	5. Your answer should be 225
6. Multiply 65 x 65 mentally. Multiply 95 x 95 mentally. Multiply 55 x 55 mentally.	6. ANSWER: 4225 ANSWER: 9025 ANSWER: 3025

This is much shorter than the programs typically used today in schools or industries, but it illustrates many of the characteristics of modern programs.* Let's compare it with the tutorial session just described.

In the first place, it begins with what a learner can do, and helps him to learn some new behavior that will be useful to him. This particular program begins with a person who knows how to multiply and teaches him to perform a particular short-cut in multiplying. In the same way I began, in the tutorial, with an ability to add and some knowledge about fractions, and learned to add fractions.

The program, like the tutor, tries to lead the student as efficientiy and directly as possible from the behavior already learned to the desired behavior. Notice how it is done. The learner is kept active. Like a

---

*A longer program to teach the same skill has been published by Teaching Materials, Inc. The author is James L. Evans. A comparison of the two programs is an interesting exercise.

student in a tutorial, he is kept busy answering questions, filling in blanks, solving problems, practicing responses he can use. Learning is an active thing. After each response, the learner is given knowledge of results -- that is, told whether his answer is the correct one. (If the tutorial student is wrong, the tutor tells him; the student of programed instruction performs the required operation, then looks at the answer column, which he usually keeps covered until he finishes the item, or, in some programs, has to turn the page to read.) The importance of receiving knowledge of results at once should not be underestimated. When a student has to wait a day or more until the teacher grades the papers, most of the learning effect of checking up on one's own performance is lost.

The program, like the tutor, gives the student a lot of help at first, then less and less until he is doing the entire operation without prompting. This technique of gradually reducing the cues is called fading or vanishing, and is an important part of most types of programing. Its goal is to make the learner finally independent of the program (or of the teacher). Notice also that the program, like the tutorial session, is adaptable to the learning speed of the student. The learner can move forward at his own best pace. Furthermore, the program has another device for adapting to student need which is worth noticing. If a student gets the idea quickly, any time after the second item he can skip ahead. This is a technique called branching. Branching makes it possible for a quick student to take a short path through a program, and for a slower student to choose a longer, easier, surer path.

Knowledge of results, fading, self-pacing, branching are among the methods used to contribute to efficient travel between what the learner can do when he begins and what it is desired that he should be able to do when he completes the program. But the design of a program includes something more basic than any of these techniques. The basis for constructing a good program is a detailed and penetrating analysis of the behavior to be learned and the sequences of behavior leading up to it. To take an example, the skill of adding fractions is based on other skills, and before those can be learned it is necessary to master certain subsidiary skills, and so forth. Thus, in a sense, the act of making a program (or, doubtless, planning an efficient tutorial session) is really an attempt to understand the structure of developing knowledge. What is the most efficient sequence of learning acts that lead to a desired pattern of behavior?

Here is a simple definition of the essentials of programed instruction:

> Programed instruction consists of a reproducible series of learning acts arranged in sequence to lead a learner efficiently from some behavior he has mastered to some new behavior which is the goal of the instruction.*

This is a broad definition. Some scholars would find it not restrictive enough, because it includes more than one kind of programing (see the

---

*For a similar definition, from which this one is in part derived, see Lumsdaine's article in the 1964 NSSE Yearbook, which is listed in the bibliography at the end of this paper. The word "reproducible" in the definition is a key one, because it distinguishes programs from tutorial sessions. The program is reproducible in a sense that neither the tutor nor the tutorial is.

section on "Different Kinds of Programs"). But in a time of rapid change and development in the art of programing it is better to be inclusive rather than exclusive.

Let me call your attention to the emphasis on learning acts in the definition. It doesn't make a great difference whether we call it programed instruction or programed learning. Some people who know a great deal about it, do prefer to call it programed learning. But the thing to remember is that a program is concerned with learning. It helps a student learn for himself. It is an active way to study. The things that are "programed" are acts of learning which every student will take by himself, but which are so arranged as to make his effort as productive as possible.

How programs are made

Because the programer is not going to be present, as the tutor is, when students are working with his program, he must make special efforts in advance. The tutor gets quick feedback from each of his students. He can tell at once whether a question is being misunderstood or is too hard. The programer has the challenge of making something so good that it will work without him.

Therefore, the first thing he has to do is to analyze very carefully what it is he wants his students to learn. What does he mean, precisely, by "being able to add fractions"? What kinds of problems, specifically, does he want the student to be able to solve? It is not sufficient to say, as some textbook writers and careless lesson planners sometimes do, that they want the student to "understand" something. At what level do they

want him to understand it? What do they want him to be able to do as a result of what he learns? Cook and Mechner* point out, by way of example, that before an electronics program can be built around such an objective as "understanding of diodes" it is necessary to define the objective in terms of behavior. The programer must decide whether, for example, he wants the student to be able to

a. Say what is a diode; or

b. Explain what a diode does in an electrical circuit; or

c. Name four different kinds of diodes, explain how each one works, draw its characteristics, and explain for what kind of application each is suitable.

Obviously, there is a vast difference among these three levels of mastery, and until the objective is defined in behavioral terms there is no way to design the necessary sequence of learning acts. Thus the making of programs has the great advantage of requiring programers to think very hard and precisely about what it is the student should be learning.

For this very reason, the making of programs encourages a rethinking of curriculum and course content. Schools and colleges that have made some of their own programs have reported this fresh look at course content as one of the most healthful aspects of programing. Some of the developing countries are now looking with great interest at programed instruction, not only because it allows them to extend good teaching but also because it

---

*In Margulies and Eigen, editors, Applied Programed Instruction (see Bibliography), p. 6.

provides a vehicle for the curriculum revision they so much need. Whereas it is not always the case that programers have given special care to the course content they are presenting, still it should be the case, and if programed instruction is not used as an ally of the great curricular revisions now under way, one of its strengths is being wasted.

The first task in making a program is, therefore, to analyze very sharply what is supposed to be learned from the program. The second is to design the series of learning acts leading up to the "terminal behavior," the behavior it is desired to teach. Making a program consists of preparing in advance a set of learning experiences as nearly as possible foolproof. The programer can't play by ear, like the tutor, or take a design for granted; he must plan and test. When he has devised the best learning plan he can, he prepares a part of the program and tests it on one student. One student. This is so that he can study very closely the student's reactions and responses. Programed instruction is very good for this purpose; the student responds to every item, and thus informs the programer at every step whether learning is or is not taking place. The writer of a textbook has to wait for a long time before he gets much student feedback, and then it comes in broad and general terms. With the programer, however, feedback on every item in the program is a part of program-making. Testing one's instructional materials in this detailed way is a remarkable, an eye-opening experience, as hundreds of teachers, exposed to the task of making a program, have testified. After observing what one student does with the embryo program, the programer revises it, and tries it on another

student. The process of revising and testing goes on until the program has been tried on, say, ten students. Then it is ready for testing on a large group.

Not all programs, of course, are made this carefully. But they should be, and many of them are. And how a program is made is very important to a teacher or a school, because a well-tested program is, by definition and by measurement, an efficient teaching instrument. When one chooses a textbook, it is possible, of course, to judge the authenticity of the content and to estimate, in a general way, what a student may be expected to learn from it. But when one chooses a program, if there is evidence that it has been well tested, then one can not only judge the content but also estimate with considerable confidence exactly what and how much a student of a given kind will learn from it.

Different kinds of programs

It is easier to understand the different types of programs if one knows a bit about how programed instruction originated.

Although in one sense programed instruction is as old as tutorial teaching -- and certainly both its theoretical and its practical roots are much older than the 1950's -- still the name "program" as applied to a sequence of learning events, and the great development of programed instruction in recent years, date only from an historic paper by Prof. B. Fred Skinner, of Harvard, in 1954. The paper was entitled "The Science of Learning and the Art of Teaching."* It grew out of Skinner's experiments

---

*This paper is readily available in Teaching Machines and Programmed Learning, edited by Lumsdaine and Glaser (see Bibliography).

with animal learning, in which he had achieved quite remarkable results by "response shaping" -- that is, by rewarding desired responses, so that the animal would be encouraged to move nearer and nearer to the behavior desired of him. Skinner tried versions of this with humans, and found that it helped them, too, to learn faster. He tried it in teaching psychology to Harvard students, and again it worked. In the latter cases, his programs were sequences of items, to each of which the student responded and then was shown the correct answer. By making the steps between items short, it was possible to ensure that most answers would be correct, and therefore that students would practice correct rather than incorrect responses, and be "rewarded" by being told they were right. And thus they would move, step by step, toward the behavior that was the goal of the program.

The great contribution of Skinner was the idea that educational subject matter "could be regarded as an accumulative repertoire of behavior which could be analyzed logically into a number of small 'steps' representing increments of successive approximation to final mastery." It follows, then, that it should be possible to design an optimum sequence of steps by analysis and trial. This is the basis of Skinnerian programing.

The great majority of all programs now being made follow the Skinnerian pattern, and are usually called <u>linear</u> programs. I have elsewhere* summed up the characteristics of a Skinnerian program thus:

---

*<u>Programed Instruction Today and Tomorrow</u> (see Bibliography).

(a) an ordered sequence of stimulus items,

(b) to each of which a student responds in some specified way,

(c) his responses being reinforced by immediate knowledge of results,

(d) so that he moves by small steps,

(e) and at his own pace,

(f) therefore making few errors and practicing mostly correct responses,

(g) from what he knows, by a process of successively closer approximation, toward what he is supposed to learn from the program.

In the 1920's, however, S. L. Pressey had already used the principles of frequent response, immediate knowledge of results, and self-pacing to devise a set of punchboards or printed tests which could be used to review expository material or to focus the learning on the important points of a lecture. These were successful in increasing the amount of learning. Inasmuch as these devices were used as adjuncts to other teaching materials, they became known as adjunct programing.

Teaching machines and teaching programs were also studied and used in U. S. military training in the 1940's and 1950's, and a number of the principal researchers on programed instruction came out of this experience. Their tendency, it should be noted, was to use Guthrie's contiguity learning theory, rather than Skinner's reinforcement theory.

One of the programers who worked for a time in military training was Norman Crowder. He developed a method sometimes called intrinsic programing.

This is essentially a bit of text followed by a multiple-choice question. If the student chooses the correct response, he is given an approving word and sent on to the next topic. If he chooses an incorrect response, he is given some explanation of why his choice was wrong, sometimes is reproved mildly, and then sent back to try again.

We can illustrate these several kinds of programing by items. The first item in the Holland-Skinner linear program on psychology* is:

> A doctor taps your knee (patellar tendon)
> with a rubber hammer to test your _____. (reflex, reflexes)

An item of the kind that might appear in a Pressey adjunct program is:

> While being driven above the speed limit on a slippery road, an automobile skids out of control and kills a pedestrian. The driver will most likely be charged with (choose one)
>
> a. First degree murder
> b. Second degree murder
> c. Manslaughter

And here is an item abstracted in part from one of Crowder's programs:**

> If we had to write out the multiplication in which 2 is used as a factor nine times, it would be rather cumbersome.
>
> 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 = 512

---

*Holland, James G., and Skinner, B. F. The Analysis of Behavior. New York: McGraw-Hill, 1961.

**Crowder, Norman A. The Arithmetic of Computers. Santa Barbara, California: USI Western Design, 1960.

For convenience, there is a shorthand way of writing a phrase such as "the product formed by using 2 as a factor nine times." The whole phrase may be written simply as "2^9". This symbol means

the number formed by
using this number ⟶ 2 ^{9} ⟵ this many times
as a factor

Thus 2^9 means (or equals)

$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 512$

Similarly

$2^3 = 2 \times 2 \times 2 = 8$

$2^4 = 2 \times 2 \times 2 \times 2 = 16$, etc.

What would the symbol 3^4 mean?

ANSWER	Page
$3^4 = 3 \times 4 = 12$	3
$3^4 = 4 \times 4 \times 4 = 64$	18
$3^4 = 3 \times 3 \times 3 \times 3 = 81$	29

The Crowder programs are typically printed in a "scrambled book," one item or answer to a page. If the student selects the first answer to this item, he turns to page 3 and reads, "You aren't paying attention" -- then reads an explanation, and is sent back to the original question. If he chooses the second answer, he goes to page 18, and sees:

> You have it backwards. The symbol 3^4 means you are to use the 3 as a factor four times. Your answer was the result reached by using 4 as a factor three times.
>
> Now return to page 10 and try again.

But if he chooses the third answer, on page 29 he reads

> You are correct. The symbol 3^4 means "the product formed by using 3 as a factor four times."

And then he is given the next item.

Programing is now rather fluid. Although most programs are in linear form, branching is being introduced into many of them to permit a student to go faster if he can. Although self-pacing is supposed to be one of the strengths of programed learning, still it has been shown that under certain conditions a program can be presented effectively on television or film where the pace is controlled by the presentation rather than the student. Experiments have been conducted using programs to teach by discovery methods rather than the conditioning which is associated with Skinner's approach. There have been combinations of linear and intrinsic programing, and of constructed responses (typical of Skinner's programs) with multiple-choice responses (typical of Pressey's and Crowder's). In other words, at this very early stage in the making of programs -- most programs have become available only since 1960 -- it is hard to tell exactly what forms they will take in the future, and there is good reason to hope that considerably more effective programs may be made than are now being produced.

What teaching material can be programed?

It seems reasonable to believe that any educational subject matter in which the terminal behavior can be specified clearly, can be taught by means of programs. We do not know exactly how wide are the boundaries within which programs can teach efficiently. Yet it is clear that some subject matter -- for example, mathematics -- readily lends itself to programing, whereas music appreciation does not. Similarly, the study of grammar can be programed, but the practice of creative writing does not lend itself to programing.

A key to the question is suggested by the kinds of programs that are being made and used. Mathematics is the most programed subject; English grammar and correctness, next. There are a number of programs in technical skills, and some in science, spelling, foreign language, and so forth. There are relatively few in the humanities -- literature, philosophy, history, and so forth. Thus, in general, the subjects where ambiguity is low, and where objectives are specific and behavioral, seem to be the ones most likely to gain through programed learning.

Do programs work well in practice?

Any program carefully made and tested can be counted on to teach effectively, to the kind of students it has been made for, the kind of behavior it has been designed to teach. The research studies on programs in classrooms and industries have, in general, been very encouraging. Students using programs have ordinarily gained at least as much as under traditional methods. Combined with classroom teaching, programs have often taught significantly more, in the same time, than either method alone.

The real problem is not whether students can learn from programs, but rather where and how programs can be used most efficiently in a package of learning experiences. This undoubtedly varies with the situation. For example, some of the developing countries, desperately short of teachers and faced with an enormous school-leaving problem because after the first few grades many students have no schools to go to, look to programs as a means by which students can learn by themselves with a minimum

of direction or supervision. There, it may be a choice of programed instruction or nothing. In the relatively well-staffed and plentiful schools of a country like the United States, on the other hand, programs have come into use rather slowly. They have been gladly accepted as "modules" in the program, to take care of such a specific objective, for example, as teaching a student to use the slide rule. They have also proved useful in a number of remedial tasks, such as helping overcome grammatical faults. Use of programs of these kinds helps release the teacher for other duties, and permits the needs of one student to be met without holding up the entire class. The dream of a curriculum in which every student, using mostly programed materials, would move forward at his own pace, is still very far from being realized. In only a handful of schools, like the Brigham Young University experimental school in Provo, Utah,* has enough of the curriculum been turned over to programs to make possible a test of what programed instruction can contribute to the individualizing of instruction. It is clear that programs themselves will not individualize instruction; a fundamental revision of teaching practices and school programs is required. Furthermore, programs are not available either in quantity or quality to make possible a test of what they can contribute to substantial portions of the curriculum. The Provo school could find reasonably satisfactory programs for a full sequence of mathematics, first grade to 12th,

---

*For a case study of the Provo experience and an idea of the problems involved in trying to give programs an important place in the curriculum, see Four Case Studies of Programed Instruction in American Schools, listed in the bibliography.

but for no other similar curricular sequence. It should also be added that a certain amount of boredom is reported in many cases, especially with long programs of the kind presently available; it seems that confirmation of results is not as powerful a reinforcement as it had been thought to be, and teachers must provide additional rewards or a greater variety of activities. Therefore, the most common practice in United States schools so far has been to work programed units into a "package" of learning experiences, using the programs typically for remedial, drill, or enrichment purposes.

It must be remembered, of course, that we are at a very early point in the history of programed instruction. When more and better programed materials become available, they may be given more important tasks within the curriculum. But at present the most common concept in schools where programs are used effectively is that certain things can be done best by the teacher, some by textbooks, some by films or television, and so forth, and some by programs. The problem is to match up the tasks with the materials.

It should be said that interest in and use of programed instruction is far wider than the United States. There is very lively activity in the United Kingdom, Germany, France, Scandinavia, the Soviet Union, and Japan, among others, and the method has been introduced by Unesco projects into Latin America, West Africa, and the Middle East.

How can you recognize a good program?

An experienced person can tell a skillfully made program from one that

is unskillfully made. A subject matter expert can judge the subject competence of a program as readily as that of a textbook. However, trying to predict from which of two programs a student will learn more, when both programs are skillfully made and both are competent in subject matter -- this is doubtful procedure. It has often been tried, and the experts are little better than the laymen in making the prediction. The only safe thing is to test the programs on students.

If a school is considering the adoption of a program in a given subject, its first step might well be to have the program looked at by a person or persons (teachers or others) who know something about the art and science of programing. If they say the program appears to be competently made, then it should be read and inspected by teachers who know the subject matter well: does it attempt to teach what needs teaching? If it passes this test, then it can be tried on a small group of students.

This third step will become less necessary when program publishers supply adequate test data with their programs.

When is a teaching machine necessary?

Manufacterers jumped into programed instruction early and began to make teaching machines. Indeed for a long time there were more machines than programs. The edge has been taken off the teaching machine market, however, and the great majority of programs now used in the United States are in the form of printed programed textbooks.

Teaching machines will be used, in the future, for tasks where machines are essential. The vast majority of machines available today are little

more than covers to hold the programed book. Cheating has not proved to be a serious problem, and therefore most of these machines contribute little to the ultimate effect of the program. In certain cases, however, one must have a machine to present a program. For example, a program for a child too young to read can be presented perfectly well with a machine, but not with a book. A foreign language program that tries to teach listening comprehension obviously requires something more than a book. The military services have found a number of teaching machines very useful in helping personnel to learn specific skills. And as teaching machines become more sophisticated, particularly as computers come to be used more and more for the purpose, then undoubtedly they will be used more, and more programs will be written for them.*

Where can one read more about it?

There are a great many books and articles on programed instruction, and we can only name a few of them here. However, among the most useful volumes are:

Teaching Machines and Programmed Learning, edited by Arthur A. Lumsdaine and Robert Glaser (Washington, D. C.: National Education Association, Division of Audio-Visual Instruction, 1960). This is a collection of

---

*A survey of teaching machines is in Teaching Machines and Programed Learning, A Survey of the Industry, prepared by James D. Finn and Donald G. Perrin (Washington, D. C.: U. S. Office of Education, 1962). See also pp. 217-80 of Margulies and Eigen, Applied Programed Instruction (see Bibliography).

papers, including the famous articles by Skinner of 1954 and 1958, articles by Pressey and Crowder, and a generally good introduction to the state of the art at the end of the 1950's.

The Learning Process and Programmed Instruction, by Edward J. Green (New York: Holt, Rinehart, and Winston, 1962) is an easy-to-read introduction to the Skinnerian approach to programing.

Applied Programed Instruction, edited by Stuart Margulies and Lewis D. Eigen (New York: Wiley, 1962). This is a collection of papers.

A Guide to Programmed Instruction, by Jerome D. Lysaught and Clarence M. Williams (New York: Wiley, 1963). This is a general introduction.

There are a number of programed introductions to programed instruction. Of the three mentioned below, the first two are in linear form, the third in intrinsic programing in the style of Crowder:

A Programed Primer on Programing, by Susan Meyer Markle, Lewis D. Eigen, and P. Kenneth Komoski (New York: The Center for Programed Instruction, Inc., 1961).

Programed Instruction: A Manual of Programing Techniques, by Dale M. Brethower (Chicago: Educational Methods, Inc., 1963).

Explaining Teaching Machines and Programming, by David Cram (San Francisco: Fearon Publishers, 1961).

Programed Instruction Today and Tomorrow, by Wilbur Schramm (New York: The Fund for the Advancement of Education, 1963) is an evaluation of the state of the art as of the end of 1962, and an attempt to relate programing to the great currents of change in education.

A useful little book on analyzing the objectives of programed instruction -- or of any other kind of teaching -- is Robert F. Mager's little book Preparing Objectives for Programmed Instruction (San Francisco: Fearon, 1962).

The 1964 Yearbook of the National Society for the Study of Education, is entitled Individualized Instruction and is edited by Ernest R. Hilgard (Chicago: the NSSE). A number of articles in the volume deal with programed learning.

Four Case Studies of Programed Instruction in American Schools, by Wilbur Schramm, Herbert Thelen, A. W. Foshay, Jack V. Edling and others (New York: The Society for the Advancement of Education, 1964) is a record of the experience of schools in four cities in introducing programed learning.

Programs '63, edited by the Center for Programed Instruction for the U. S. Office of Education (Washington, D. C.: Office of Education, 1963) contains brief descriptions and page-long samples of all programs known to have been available in the United States at the beginning of 1963.

Abstracts of all the research on programed instruction up to about the beginning of 1963 are to be found in a small book, The Research on Programed Instruction, prepared by Wilbur Schramm and published by the U. S. Office of Education (Washington, D. C., 1964).